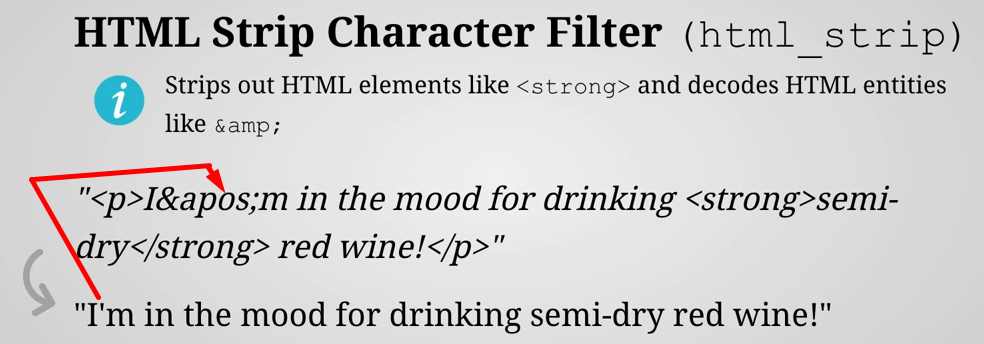
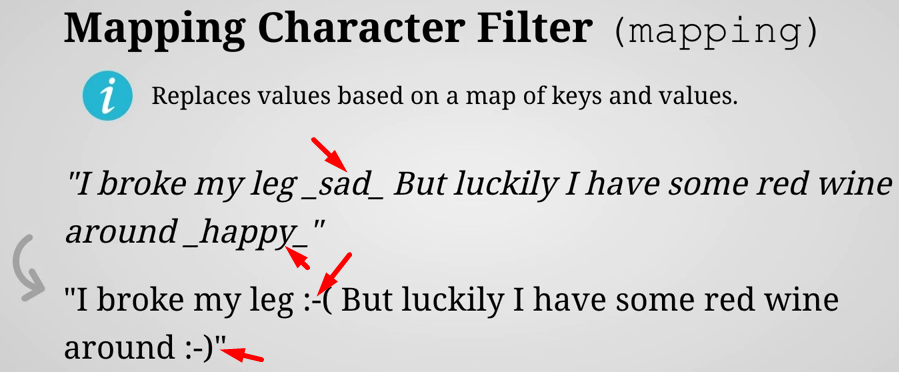
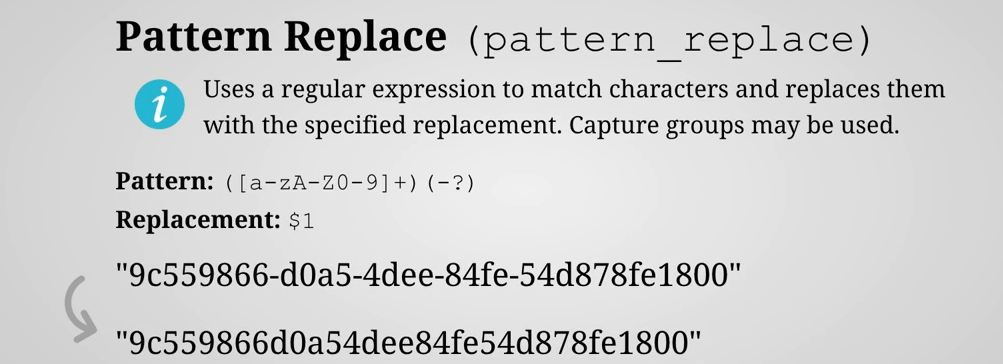
1. 
2. Now we know what is analyzer (character filters, tokenizer, token filters).
3. Let’s see which are available by **default**.
4. **Agenda**:
5. If you know how to use one tokenizer for instance, using another one is no different.
6. Let’s begin with an overview of the build-in character filters.
7. There are only 3 character filters available with the most important one being the **HTML Strip Character Filter(html\_strip)**.
8. **html\_strip**:
   1. This filter is used to strip out the html from text but it also decodes the html entities (&amp;).
   2. **Real Example**: You’re indexing comments from a blog and that may contain HTML markup or perhaps you have a number of blog posts that contain html markup from a text editor so called “What You See Is What You Get”.
   3. Since the HTML would not be of any use for us in regards to searching we might as well get rid of the markup before adding documents to the index.
   4. **Example**: In the example that you see now you can see how the HTML text have been removed but also how the apostrophe HTML entity has been converted into an actual apostrophe.  
      
9. **Mapping Character Filter**(mapping)
   1. The mapping filter replaces values based on a supplied list of values and their replacements.
   2. You could define that certain characters should be replaced by smiley faces as in this example.
   3. You simply specify a map of **tokens** and their **replacements** and the character filter will take care of replacing all of the occurrences with the appropriate replacements.  
      
10. **Pattern Replace** (pattern\_replace)
    1. Same filter except that it does the matching based on **regular expressions**.
    2. It also allows you to access the matched values with **capture groups** and use those with the new replacement values.
    3. **Example**:
       1. In this example I have to find a regular expression that matches a series of numbers potentially followed by hypen(-).  
          This expression includes a capture group which I refer to as dollar sign one with the replacement value. Instead of that I could just define a plane value if I wanted to.  
          So, the result of this would be that the hyphens are removed although that could certainly be simplified.
    4. ****